MANAGEMENT OF INFORMATION FOR MISSION OPERATIONS USING AUTOMATED KEYWORD REFERENCING

Roger A. Davidson and Patrick S. Curran N94-23886

Space Flight Operations Section
Jet Propulsion Laboratory, 301-345
California Institute of Technology
Pasadena, California 91109-8099

ABSTRACT

Although millions of dollars have helped to improve the operability and technology of ground data systems for mission operations, almost all mission documentation remains bound in printed volumes. This form of documentation is difficult and time-consuming to use, may be out-of-date, and is usually not cross-referenced with other related volumes of mission documentation. A more effective, automated method of mission information access is needed. In our paper we propose a new method of information management for mission operations using automated keyword referencing. We expound on the justification for and the objectives of this concept. We will share the results of a prototype tool for mission information access that uses a hypertext-like user interface and existing mission documentation. Finally, we will describe the future directions and benefits of our proposed work.

Key Words: Hypermedia, hypertext, information retrieval, mission control, operations, automation

1. INTRODUCTION

Mission complexity, longevity and organizational size have made the documentation of mission information increasingly more important. Unfortunately, documentation is expensive. It must be prepared and distributed. Often, people must be trained in how the hoards of documents are constructed and used and where they can be found. Then, of course, there are corrections and updates to documents which result in new versions that must be distributed. This says nothing about the configuration control and security issues involved with creation of and access to mission documentation. Moreover, even with all its associated costs and headaches, documentation is worthless unless it can be used effectively. In fact, out-of-date documentation can be misleading, resulting in increased operating cost and risk. One can begin to see why "the good ol' days" of space exploration (when dedicated teams of experts needed less documentation) appear so pleasingly simple.

An example illustrates some of the burdens that hardcopy documentation places on mission operations activities. The Command Dictionary, Telemetry Dictionary, and the Flight and Mission Rules Dictionaries are essential references for the mission operations functions of mission control, command verification, anomaly investigation and training. However, these documents are typically bound separately (having been created by different organizational teams), and the cross-referencing of these related information sets is often found scattered in the textual description of the entries. Therefore an analyst investigating an anomaly on a specific telemetry channel must search for its entry in the telemetry dictionary, read to find related telemetry channels, open the

command dictionary to investigate the commands that affect the telemetry channel, and try to cross-reference with the flight and mission rules to understand the related commanding and scheduling constraints. The resulting, human-generated flurry of bookmarkers and Post-it Notes™ is a frustrating, costly and unproductive use of an analyst's time.

1.1 Use of Milo for Galileo

The Jet Propulsion Laboratory's Galileo Project took a significant step towards solving these problems by making their telemetry and command dictionaries and their flight and mission rules available in an electronic form to their mission control and spacecraft teams. In late 1989, an Operations Engineering Team (OET) Systems Engineer named Mark MacDonald wrote a program script in a commercial text editor for personal computers. The program, named Milo, offered only a command line interface, was slow, did not show document graphics, and was not accessible from the principal computer system used by the OET and Mission Control Team (MCT). Even with these negatives, it was observed that OET and MCT personnel were more likely to look for information with Milo than search through piles of printed documents.

1.2 MMIT Prototype

In 1991, JPL's Space Flight Operations Section was funded by the Advanced Systems Office, of the (then) Flight Projects Support Office (now called the Multimission Operations Systems Office), to build a prototype "User Environment for Multimission Control" (Ref. 1). A goal of this work was to prove a new concept called Object-Oriented Operations (O^3) in which data objects, representing basic units of mission information such as telemetry channel data, can be shared among different workstation applications using a graphical "drag-and-drop" user interface paradigm (Ref. 2). For example, a user could be editing a decommutation map with one software tool, "grab" a telemetry channel object then "drag" the object over to an electronic reference tool and "drop" it; at which point the reference tool would display the available mission documentation on that channel.

The approach was to construct three software prototypes that could interact as just described. The chosen tools were a telemetry channel plotting tool, a graphical decommutation map editor, and an electronic reference tool. That reference tool, called the Multi-Mission Information Tool (MMIT), was designed to duplicate all the current capabilities of Galileo's Milo, and incorporate the technologies of graphical user interfaces and hypermedia.

2. CAPABILITIES OF MMIT

MMIT is written in C with the XView™ library and runs in the UNIX environment with the X Window System™. The program can be compiled to run on any Sun™ workstation or compatible. The following paragraphs describe the important characteristics of the tool.

2.1 Graphical User Interface

MMIT presents a graphical interface to its users. From top to bottom, the MMIT interface contains a menu bar, a scrollable text window, and three scrollable lists and a message area (side by side) (Fig. 1). The menu bar is minimal and standard. The text window is used to display mission document pages.

The three lists and the message area display information that was not available to Milo's users. The first list shows the dictionaries (or document sets) which are available and cross-referenced for access by MMIT. The second list shows either a list of dictionary entries (if no particular entry has been selected and displayed),

or it shows the cross-referenced entries, of each dictionary, to the entry shown in the text window. The third list displays a history of the session by listing the names of all prior entries selected by the user. The message area is used to notify the user of errors or other system messages.

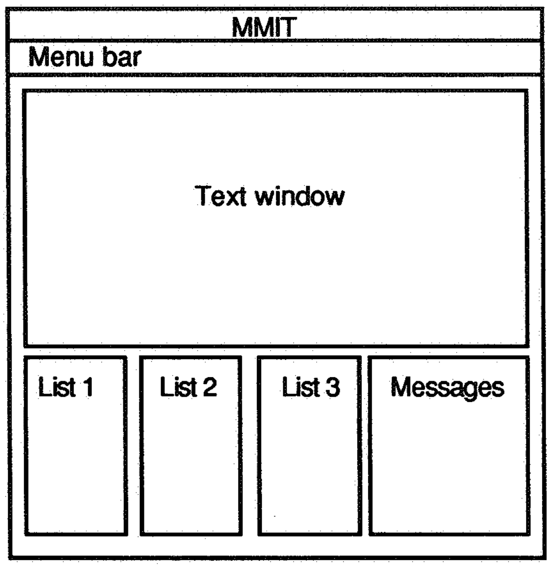

Figure 1. Layout of MMIT User Interface

2.2 Use of Standard Text Files

A primary design goal of MMIT was to permit the use of existing, unmodified document files. The MMIT prototype uses text files that were electronically transferred from the Milo system to a workstation. The only modifications to the files were simple removals of word-processor-specific formatting characters, and this process was automated using scripts.

2.3 Simple Cross-referencing

Because mission documentation is interrelated and its use is often centered on this relational nature, the provision of a user-friendly cross-referencing capability in MMIT is a primary feature of the tool. However, because we wanted to make as few modifications to the document files as possible, the nominal hypertext approach, placing hypertext markers within the files, was not desired.

Instead we chose to construct indices into the documentation which MMIT would read and then display in its scrollable lists. This task was significant because the prototype included nearly 3000 document files. We were able to automate the task because the names of the index entries that were referenced in the documentation were the same as the filenames containing the information on that entry. We let the computer search through the documentation and build our cross-reference table for us.

2.4 Expandability

Because the document files are left unchanged, and because the cross-reference indices are built using automated scripts, MMIT is fully expandable to encompass new sets of related mission documentation. The only required modification would be to the MMIT initialization files which contain the cross-reference indices, and this is an automated process.

2.5 Object-oriented

MMIT uses object-oriented programming techniques. We believe that this practice results in software with higher reusability, modularity, maintainability, and reliability.

2.6 User Evaluation

User evaluation was an important part of the MMIT prototype development process. User comments helped to refine the user interface and correct problems with the basic capabilities. During the prototyping work, MMIT was evaluated by members of the Galileo MCT who guided the development such that they now use the prototype in their daily operations.

Since that time, MMIT has also been demonstrated to members of the Multimission Control Team and the Mars Observer Spacecraft Team. These interactions have enabled us to construct a list of capabilities for a new version of MMIT that will be even more useful and cost effective in operations.

3. PLANNED CAPABILITIES

3.1 Motif Compliance

The most visible change to the MMIT user interface is the planned modification to comply with the OSF/Motif™ window manager and style guidelines. Motif is the standard windowing style for JPL's telemetry system, the Multimission Ground Data System (MGDS). Since MMIT will interface directly with MGDS and have the same users as MGDS in the operational setting, the transition to Motif is a logical and significant step towards making the tool more usable.

3.2 Decommutation Map Generation

On board the spacecraft, data from the many sensors and instruments are assembled in a structured order before the telemetry stream is radiated back to Earth. This process is called commutation. Once the telemetry is received, the data stream is decommutated back into its original elements, called *channels*.

In MGDS, the process of telemetry decommutation is guided by a software decommutation map, or decom map, which specifies where each channel occurs in the telemetry stream and how many bits the channels contain. The decom maps are written in a high-level, MGDS-specific language called DMDL. The decom map files are long and complex since there are typically thousands of spacecraft channels and the process of decommutation varies with bit-rate, mode of spacecraft operation and mission phase. The process of editing decom maps is prone to human error, and yet correct decom maps are critical to the interpretation of realtime telemetry.

A future capability of MMIT will allow users to generate and update decom maps directly from the telemetry channel documentation. With proper change control, we feel this can greatly improve the reliability and efficiency of the decom process.

3.2 CPT Generation

The MGDS system uses a Channel Parameter Table (CPT) to group telemetry channels according to their relevant spacecraft subsystems, to assign channel types, to set channel alarm conditions, to specify data number to engineering unit conversions, and invoke more involved channel processing. The CPT is created and updated from channel definition commands and is then converted to a binary form that is read by MGDS on start up (much faster than ASCII format).

It is not unusual for the CPT to change on a daily basis because channel alarm conditions constantly change over the course of a mission. The generation of a binary CPT is an involved one, so generally the MCT uses a binary CPT as a baseline with a supporting ASCII file of channel definition commands read in as changes from the baseline. This supporting file grows as new updates to the CPT are added. On a regular basis, the supporting file is integrated with the binary CPT and the procedure begins again.

This process could be entirely automated by generating the CPT directly from mission documentation. This type of automation has been used for the Deep Space Network (DSN) monitoring channels (Ref. 3). We plan that the next version of MMIT will be capable of such automation for all the spacecraft channels as well.

3.4 Anomaly Diagnosis Support

MMIT can also be used to support anomaly diagnosis, with relatively few modifications. By including existing anomaly reports in the supported MMIT mission document set, users would be able to cross-reference channel and command information with previously reported anomalies. In a realtime setting, this could speed initial diagnosis for recurring anomalies. For non-realtime anomaly isolation, such a reference tool could reveal patterns of anomalies and report previous anomaly resolutions in a quick and efficient manner. This capability would also allow projects to share ground system anomaly reports, and thereby increase the diagnostic reference capacity of the tool. Finally, it would be a significant step towards electronic anomaly reporting and processing.

3.5 Sequence Checking Support

Flight rules are operational constraints for spacecraft. MMIT already includes the flight rule documentation as part of its mission document set. Work is in progress at JPL on a syntax that specifies "flight rules" for command sequence checking programs (Ref. 4). Future versions of MMIT may be able to use this syntax to serve as an electronic reference for sequence checkers.

4. LIMITATIONS AND SOLUTIONS

From user evaluations and previous design decisions, we understand that MMIT has some limitations currently. As more capabilities are added to the tool, we can see other possible limitations rising to greet us. Here are our most significant concerns.

4.1 Indexing with Filenames

Currently, the mechanism for cross-referencing documents in MMIT is based on initialization files that are read at application start up. These files are generated in an automated manner using scripts outside the MMIT program. As previously mentioned, this automation is possible because the document filenames are derived from the document entry name. As an example, the file containing the information on channel e-1004 would probably be named, e1004.tlm.

Although this is a legitimate design for MMIT because mission documentation has this structure, the applicability of MMIT to other groups of documentation where this structure did not exist would be limited.

4.2 Cross-reference Automation

The method of cross-reference automation used by MMIT requires a great deal of compute time to construct the cross-reference file. At run-time, however, this design allows the user to jump between related documents quickly by clicking with the mouse on the provided list of related documents. This results in a hypertext interface that allows linking of related documents without modifications to the documents themselves.

Although the design is an effective engineering solution, it does have limitations (Ref. 5). The hypertext links are based only on the internal references in the document. In other words, a user could be shown documents related to a single channel, but the user could not see a list of channels in the same subsystem as that single channel. Further, when the related document is displayed, the reference is not located or highlighted. However, the use of highlighting is planned.

In order to support new types of links, a cross-reference table would have to be generated for each type. With the large number of mission documents supported, run-time maintenance of the cross-reference table would result in unacceptable performance. This

limits the complexity of searches in MMIT to simple word searches and to the cross-reference tables already generated. Fortunately, because the type of documents supported and the typical uses for MMIT are known, cross-reference tables can be constructed in advance to support the large majority of queries.

4.3 Response Time

The large number of mission documents can result in unacceptable response times to users. So far, we have been able to compensate for this with design decisions such as the use of cross-reference tables over dynamic searches each time a document is displayed.

In the future, MMIT may become performance limited as new types of documentation (such as the anomaly reports) are added to the mission document set, or as the new capabilities of CPT and decom map generation are added. Almost certainly, CPT and decom map generation could become separate processes which are started by MMIT and which notify the user when complete. We feel the current relative growth in computing power will keep pace with (if not exceed) the relative increase in mission documentation.

4.4 Configuration Control

As MMIT becomes more integrated with critical operations tasks like CPT and decom map generation, the control of changes to the mission documentation becomes more important. To support critical operations requirements while allowing mission information to be distributed without fear of unauthorized changes, we propose to develop MMIT in two forms, an editor and a viewer. The MMIT Editor would only be supplied to cognizant individuals with the authority to change the documentation (so that the MCT could modify channel alarm limits for instance). The MMIT Viewer would allow a user to reference and display all mission documentation, just like the Editor, but without the capability to alter the information viewed.

5. ACKNOWLEDGMENTS

This work was done at the Jet Propulsion Laboratory, California Institute of Technology, under a contract from the National Aeronautics and Space Administration. We would like to acknowledge the work of the technical staff in the OEL and the JPL Mission Operations Teams for their enthusiasm and support. We would like to especially thank John Louie for his exceptional technical support of the MMIT prototype development.

6. REFERENCES

1. Flight Projects Support Office. 1991. *Advanced Systems Office Technology Initiatives Annual Report*, JPL internal document D-9074, 25-30.

2. Murphy, S., Miller, K. and Louie, J. 1992. "Object Oriented Operations (O^3) for Mission Control and Data Management", to be published, *Space Ops '92 Conf.*

3. Hurley, Daniel and Curran, Patrick. 1992. *Multimission Monitor Channel Dictionary*, JPL internal document D-9595.

4. Alkalaj, Leon. 1992. *Towards a Specification Language and Programming Environment for Concurrent Constraint Validation of Spacecraft Commands*, JPL internal document D-9921.

5. Glushko, Robert J. 1992. *Successful Hypertext Projects*, ACM Conference on Human Factors in Computing Systems, Chi '92, conference tutorial, 98-112.